

AI-200T00: DEVELOP AI CLOUD SOLUTIONS ON AZURE



DURATION	LEVEL	TECHNOLOGY	DELIVERY METHOD	TRAINING CREDITS
5 Days	Intermediate	Azure	Instructor-led / VILT	N/A

INTRODUCTION

This course teaches developers how to create, monitor, and troubleshoot AI solutions on Microsoft Azure. Students will learn how to implement Azure compute and containerization patterns to host applications, build serverless APIs with Azure Functions, and integrate services using event-driven and message-based architectures such as Azure Service Bus and Event Grid. The course also covers working with Azure data services that support AI workloads, including designing and querying solutions with Cosmos DB for NoSQL, Azure Database for PostgreSQL with pgvector, and Azure Managed Redis for caching, streaming, and vector search. By the end of the course, developers will be able to connect services, orchestrate AI workflows, and build secure, scalable, and observable AI-driven applications on Azure.

AUDIENCE PROFILE

This course is designed for developers who build backend and AI-driven applications on Azure and need practical skills in containerized compute, data services for AI, event-driven workflows, and application security and monitoring.

PREREQUISITES

- Programming experience with languages such as Python, JavaScript, or C#.
- Basic understanding of Azure services and cloud computing concepts.
- Familiarity with containerization fundamentals.
- Access to Azure Kubernetes Service and related tools for practical exercises.
- Understanding of machine learning concepts including embeddings and similarity search.

COURSE OBJECTIVES

After completing this course, students will be able to:

- Implement container application hosting on Azure
- Deploy and manage apps on Azure Container Apps
- Deploy and monitor applications on Azure Kubernetes Service
- Develop AI solutions with Azure Cosmos DB for NoSQL
- Develop AI solutions with Azure Database for PostgreSQL

COURSE CONTENT

Module 1: Store and manage containers in Azure Container Registry

Use Azure Container Registry to store, build, and manage container images for AI applications. Learn the registry hierarchy, build images with ACR Tasks, and implement tagging strategies for reliable deployments.

Lessons:

- Introduction
- Registries, repositories, and artifacts

- Build and run images with ACR Tasks
- Tag and version images
- Exercise - Build and manage a container image with ACR Tasks
- Module assessment
- Summary

After completing this module, students will be able to:

- Explain how Azure Container Registry organizes images using registries, repositories, and artifacts

- Build and manage container images in the cloud using ACR Tasks
- Implement tagging and versioning strategies for reliable container deployments
- Use the Azure CLI to manage container images and run ACR quick tasks

Module 2: Deploy containers to Azure App Service

Deploy and manage containerized applications on Azure App Service by configuring container sources,

runtime settings, application configuration, and diagnostics.

Lessons:

- Introduction
- Deploy containers to Azure App Service
- Configure container runtime behavior
- Configure application settings
- Observe and troubleshoot containerized apps
- Exercise - Deploy a container to Azure App Service
- Module assessment
- Summary

After completing this module, students will be able to:

- Deploy custom containers to Azure App Service from container registries.
- Configure container runtime behavior including startup commands, port settings, and persistent storage.
- Configure application settings and connection strings in App Service.
- Observe and troubleshoot containerized applications using App Service diagnostic tools.

Module 3: Deploy containers to Azure Container Apps

Create and manage container deployments in Azure Container Apps. Work with environments, runtime configuration, registry authentication, and deployment verification techniques.

Lessons:

- Introduction
- Explore Container Apps environments
- Deploy a container app using the Azure CLI and YAML
- Configure runtime settings with environment variables and secrets
- Configure image pull authentication for private registries
- Verify deployments with logs and status
- Exercise - Deploy a containerized backend API to Container Apps
- Module assessment
- Summary

After completing this module, students will be able to:

- Explain how Azure Container Apps environments affect networking, logging, and isolation.
- Deploy a container app using the Azure CLI and using the Azure CLI with a YAML definition.

- Configure runtime settings using environment variables and secrets.
- Configure image pull authentication for private registries.
- Verify container app health using logs, revisions, and replica status.

Module 4: Manage containers in Azure Container Apps

Manage container apps across the day-two lifecycle. Update images, manage revisions, diagnose failing deployments, tune resources and scaling, and troubleshoot with logs and health probes.

Lessons:

- Introduction
- Update images and manage revisions safely
- Manage the container app lifecycle
- Monitor logs and troubleshoot issues
- Configure health probes and troubleshoot failures
- Optimize container resources and scaling
- Exercise - Diagnose and fix a failing deployment
- Module assessment
- Summary

After completing this module, students will be able to:

- Update container images and manage revisions safely during development and release cycles.
- Perform app lifecycle operations and diagnose failing revisions.
- Monitor logs and troubleshoot common revision and runtime issues.
- Configure health probes and troubleshoot probe failures.
- Optimize container resources and scaling settings to balance cost and performance.

Module 5: Scale containers in Azure Container Apps

Learn how to configure automatic horizontal scaling for containerized applications in Azure Container Apps. Configure HTTP, TCP, CPU, and memory scale rules, implement event-driven scaling with KEDA, optimize compute resources, and apply revision modes for traffic management.

Lessons:

- Introduction
- Configure scale rules
- Implement event-driven scaling with KEDA
- Apply KEDA scalers for custom workloads
- Select compute resources for performance and cost

- Choose and apply revision modes
- Exercise - Configure autoscaling using KEDA triggers
- Module assessment
- Summary

After completing this module, you will be able to:

- Configure HTTP, TCP, CPU, and memory scale rules for container apps
- Implement event-driven scaling using KEDA scalers for Azure services
- Select appropriate compute resources to optimize performance and cost
- Apply revision modes to control scaling behavior and traffic distribution.

Module 6: Deploy applications to Azure Kubernetes Service

Learn how to deploy applications to Azure Kubernetes Service. This module covers creating deployment manifests, exposing applications with services, and deploying to Azure Kubernetes Service.

Lessons

- Introduction
- Create Kubernetes deployment manifests
- Expose applications in Azure Kubernetes Services
- Deploy applications to Azure Kubernetes Services
- Exercise - Deploy an AI inference API to Azure Kubernetes Service
- Module assessment
- Summary

After completing this module, you will be able to:

- Create Kubernetes deployment manifests
- Deploy and expose applications to Azure Kubernetes Service

Module 7: Configure applications on Azure Kubernetes Service

Learn to externalize configuration, secure sensitive settings, and attach persistent storage using Kubernetes features on Azure Kubernetes Service.

Lessons

- Introduction
- Define ConfigMaps for application settings
- Implement secrets for sensitive data
- Attach persistent storage to an app
- Exercise - Configure apps on Azure Kubernetes Service
- Module assessment
- Summary

After completing this module, you will be able to:

- Explain how to externalize app configuration using Kubernetes primitives.
- Implement ConfigMaps and inject settings into Pods.
- Implement Secrets and consume sensitive values securely.
- Attach persistent storage using PersistentVolume and PersistentVolumeClaim.

Module 8: Monitor and troubleshoot applications on Azure Kubernetes Service

Learn to monitor application health, inspect logs and metrics, troubleshoot pods and Services, and verify connectivity for AI workloads on Azure Kubernetes Service (AKS).

Lessons

- Introduction
- Monitor application logs and metrics
- Troubleshoot pods and services
- Verify service connectivity and endpoints
- Exercise - Troubleshoot apps on Azure Kubernetes Service
- Module assessment
- Summary

After completing this module, you will be able to:

- Explain key monitoring signals for AI workloads running on Azure Kubernetes Service.
- Use kubectl and Azure tools to inspect logs and metrics for applications.
- Troubleshoot pod and Service issues that affect AI APIs and workers.
- Verify Service and ingress connectivity paths to ensure clients can reach AI services.

Module 9: Build queries for Azure Cosmos DB for NoSQL

Learn how to connect to Azure Cosmos DB for NoSQL using the SDK, perform data operations on items, and write efficient SQL queries to retrieve document data for AI applications.

Lessons

- Introduction
- Explore Azure Cosmos DB for NoSQL
- Implement the Azure Cosmos DB for NoSQL SDK
- Query Azure Cosmos DB for NoSQL
- Exercise - Build a RAG document store on Azure Cosmos DB for NoSQL
- Module assessment
- Summary

After completing this module, you will be able to:

- Explain the Azure Cosmos DB for NoSQL resource model

and how databases, containers, and items relate to each other

- Implement SDK operations to connect to Azure Cosmos DB and perform CRUD operations on items
- Select between point reads and queries based on performance requirements and access patterns
- Build queries using SQL syntax to filter, project, and retrieve data from containers

Module 10: Implement vector search on Azure Cosmos DB for NoSQL

Learn how to store vector embeddings, execute similarity queries using the VectorDistance function, combine vector search with metadata filters and hybrid search, and use the change feed to keep embeddings synchronized.

Lessons

- Introduction
- Store and retrieve embeddings in Azure Cosmos DB
- Execute vector similarity queries for semantic search
- Combine vector similarity results with metadata filtering
- Use the change feed to trigger embedding refresh
- Exercise - Build a semantic search application with Azure Cosmos DB for NoSQL
- Module assessment
- Summary

After completing this module, you will be able to:

- Store and retrieve vector embeddings in Azure Cosmos DB containers with properly configured vector policies
- Execute vector similarity queries using the VectorDistance function to find semantically similar documents
- Combine vector search with metadata filters and full-text search using hybrid queries
- Implement change feed processing to automatically refresh embeddings when source documents change

Module 11: Optimize query performance for Azure Cosmos DB for NoSQL

Learn how to optimize query performance by analyzing query patterns, configuring range and composite indexes, selecting vector index types, and choosing consistency levels that balance freshness with cost efficiency.

Lessons

- Introduction
- Understand indexes in Azure Cosmos DB

- Configure range and composite indexes
- Tune vector indexes for embedding workloads
- Reduce RU costs with strategic indexing
- Choose consistency levels for optimal performance
- Exercise - Optimize query performance with vector indexes on Azure Cosmos DB for NoSQL
- Module assessment
- Summary

After completing this module, you will be able to:

- Analyze query patterns and use metrics to identify performance bottlenecks and missing indexes
- Configure range and composite indexes to optimize filter and sort operations for AI retrieval patterns
- Select the appropriate vector index type based on dataset size, accuracy requirements, and performance goals
- Design indexing policies that balance read performance against write costs for your workload profile
- Choose consistency levels that meet application requirements while minimizing RU consumption

Module 12: Build and query with Azure Database for PostgreSQL

Learn how to use Azure Database for PostgreSQL to build data foundations for AI applications. Design schemas, write efficient queries, and integrate with Python applications using secure authentication.

Lessons

- Introduction
- Explore Azure Database for PostgreSQL
- Connect to PostgreSQL
- Create and manage schemas
- Query data
- Integrate SDKs and applications
- Exercise - Build an agent tool backend on Azure Database for PostgreSQL
- Module assessment
- Summary

After completing this module, you will be able to:

- Explain the architecture and key features of Azure Database for PostgreSQL
- Establish secure connections to PostgreSQL using Microsoft Entra authentication and TLS
- Create and manage database schemas including tables, indexes, and constraints
- Write efficient SQL queries for common data operations

- Integrate Azure Database for PostgreSQL into applications using Python

Module 13: Implement vector search with Azure Database for PostgreSQL

Learn how to implement vector search using the pgvector extension in Azure Database for PostgreSQL. Store embeddings, create vector indexes, and build semantic retrieval patterns for AI applications.

Lessons

- Introduction
- Store and query embeddings with pgvector
- Perform fast vector similarity search
- Manage index lifecycle and embedding updates
- Run vector similarity search for semantic retrieval
- Implement retrieval patterns for RAG pipelines
- Exercise - Implement vector search on Azure Database for PostgreSQL
- Module assessment
- Summary

After completing this module, you will be able to:

- Store and query vector embeddings using the pgvector extension in Azure Database for PostgreSQL
- Execute vector similarity searches using different distance metrics and operators
- Create and manage vector indexes to optimize search performance
- Implement embedding update and refresh strategies for evolving datasets
- Build retrieval patterns that integrate PostgreSQL vector search with RAG pipelines

Module 14: Optimize vector search in Azure Database for PostgreSQL

Learn how to optimize vector search performance in Azure Database for PostgreSQL using pgvector. Tune configuration parameters, select and configure vector indexes, design efficient data layouts, scale for high-volume workloads, and implement connection pooling for AI applications.

Lessons

- Introduction
- Tune PostgreSQL for pgvector

- Choose and configure vector indexes
- Optimize data layout
- Scale for high-volume workloads
- Connection optimization
- Exercise - Optimize vector search performance in Azure Database for PostgreSQL
- Module assessment
- Summary

After completing this module, you will be able to:

- Tune PostgreSQL and pgvector configuration parameters to optimize query latency and memory usage for AI workloads
- Select and configure the appropriate vector index type based on dataset size, query patterns, and accuracy requirements
- Design data layouts that optimize vector storage and metadata filtering performance
- Scale Azure Database for PostgreSQL to handle high-volume vector workloads
- Implement connection pooling and session management strategies for AI applications

ASSOCIATED CERTIFICATIONS & EXAM

This learning path helps prepare you for the AI-200 Exam - Microsoft's Azure AI Cloud Developer Associate certification.